UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

MÉMOIRE D'ÉTUDIANT

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

CRT-619

ÉTUDE DES NOUVELLES AVENUES DE RECHERCHE PROPOSÉES EN OPTIMISATION COMBINATOIRE

par

Patrick Soriano*

* Étudiant de doctorat, département d'Informatique et de Recherche opérationnelle et Centre de recherche sur les transports, Université de Montréal.

Université de Montréal
Centre de recherche sur les transports - Publication #619
Mars 1989

RÉSUMÉ

Le développement de nouvelles approches de résolution pour traiter les problèmes d'optimisation combinatoire est un domaine qui suscite beaucoup d'intérêt depuis quelques années. Ceci provient du faite que plusieurs de ces problèmes ont des applications pratiques particulièrement importantes, mais que, malheureusement, pour bon nombre d'entre-eux,on ne connait pas d'algorithme polynomial permettant de les résoudre de manière exacte; ce sont des problèmes NP-difficiles.

Dans ce travail, nous étudions un certain nombre de "nouvelles" stratégies de résolution qui ont été proposées récemment pour traiter ces problèmes. Celles-ci peuvent se regrouper en familles distinctes selon la façon qu'elles abordent le problème des optima locaux: les "algorithmes probabilistes contrôlés" qui incluent le "recuit simulé", les "stratégies d'apprentissage" et la "recherche avec tabous".

Pour chacune de ces approches nous procéderons d'abord à une rapide description de la méthode, puis nous expliquerons les justifications intuitives ou analytiques qui la sous-tendent. Ensuite, nous décrirons ses applications à des problèmes d'optimisation combinatoire classiques tout en indiquant les résultats obtenus par rapport à des méthodes plus conventionnelles. Enfin, cette étude se terminera par une brève critique synthétique de ces nouvelles avenues de recherche.

Mots-clés: algorithmes probabilistes contrôlés; optimisation combinatoire; recherche avec tabous; recuit simulé; stratégies d'apprentissage.

1 - INTRODUCTION

Le développement de nouvelles approches de résolution pour traiter les problèmes d'optimisation combinatoire est un domaine qui suscite beaucoup d'intérêt depuis quelques années. Il suffit pour s'en convaincre de remarquer le nombre impressionnant de publications parues depuis six ans au sujet d'une de ces approches: le "recuit simulé".

Pourquoi un tel engouement ? La réponse semble tenir à plusieurs facteurs. Tout d'abord, plusieurs des problèmes que l'on retrouve sous cette appellation ont des applications pratiques particulièrement importantes et ce dans des domaines très divers. On peut citer comme exemples la gestion et la planification des transports et des ressources humaines, mais aussi la conception des ordinateurs, le design de systèmes VLSI et d'autres. Malheureusement nombre d'entre eux, dont le célèbre problème du voyageur de commerce pour ne mentionner que lui, sont des problèmes reconnus comme étant "difficiles" (appartenant à la classe des problèmes NP-complets). Ils ont comme particularité qu'on ne connaît jusqu'à ce jour aucun algorithme polynomial permettant de les résoudre de manière exacte, c'est-à-dire que pour tous les algorithmes connus le temps de calcul requis dans le "plus mauvais cas" croît au moins exponentiellement avec la taille de l'exemple considéré [1]. Or les exemples de ces problèmes que l'on rencontre dans les situations réelles possèdent souvent des dimensions telles que le temps de calcul nécessaire pour obtenir une solution exacte est absolument prohibitif.

Pour faire face à ce dilemme plusieurs alternatives ont été proposées au cours des trente dernières années. Une première option consiste à utiliser des méthodes de séparation et évaluation progressive (en anglais "branch and bound") ou des méthodes de coupes [1,2]. Bien qu'elles ne soient pas polynomiales, ces méthodes permettent d'atténuer les effets de l'explosion combinatoire [3]. En particulier des développements récents reliés à "l'étude polyédrale" des problèmes d'optimisation combinatoire ont permis d'obtenir des résultats encourageants grâce à l'utilisation conjointe de ces deux méthodes [3,4]. Toutefois la dimension des problèmes résolus et les temps de calcul nécessités sont encore loin des exigences pratiques. De plus ces techniques pour être efficaces doivent s'appuyer fortement sur la structure des problèmes étudiés et ne sont donc généralement pas applicables à d'autres types de problèmes que celui pour lequel elles ont été développées.

Une approche différente consiste à utiliser des algorithmes "approximatifs" (appelés encore heuristiques) à la place des méthodes exactes. Ces algorithmes plus simples requièrent bien moins de ressources, en nous amenant toujours à une solution réalisable. On peut alors obtenir des solutions pour des problèmes ayant des tailles bien supérieures, mais nous n'avons plus aucune garantie concernant l'optimalité de ces solutions. En effet, bon nombre des heuristiques classiques de l'optimisation combinatoire font appel à des techniques d'amélioration itérative, comme par exemple l'exploration locale, or une des limitations notoires de ces techniques réside dans leur incapacité à se dégager des optima locaux [1,3].

C'est pour tenter d'atténuer ce défaut des approches heuristiques, sans sacrifier trop de leur rapidité, qu'ont été proposées récemment un certain nombre de "nouvelles" stratégies de résolution. Celles-ci peuvent se regrouper en familles différentes selon la manière qu'elles abordent le problème des optima locaux [5]. Dans le cadre de ce travail, nous présenterons trois de ces familles, soient les **algorithmes probabilistes contrôlés**, les **stratégies d'apprentissage** et la **recherche avec tabous**. La première de ces familles fait appel à des techniques probabilistes ou statistiques pour introduire une composante aléatoire au processus de résolution. La deuxième se base sur l'accumulation d'informations pour adapter le processus de résolution au problème, au fur et à mesure que celui-ci s'effectue. Enfin, la troisième approche consiste à interdire provisoirement certains "mouvements" afin d'empêcher le cyclage de l'algorithme de résolution lorsque celui-ci "sort" d'un optimum local.

Pour chacune de ces approches, nous procéderons d'abord à une rapide description de la méthode, puis nous expliquerons les justifications intuitives ou analytiques qui la sous-tendent, ensuite nous décrirons ses applications à des problèmes d'optimisation combinatoire tout en indiquant les résultats obtenus par rapport à des méthodes plus conventionnelles. Cette étude se terminera par une brève critique synthétique.

Notons que, dans un but de concision, nous avons décidé de ne pas revenir, dans le cadre de cette étude, sur les concepts de base du domaine tels: les définitions d'un problème d'optimisation combinatoire, d'un algorithme d'amélioration itérative (et de ce que l'on entend par

"voisinage" dans ce contexte) ... Pour obtenir de plus amples détails sur ces sujets, nous renvoyons le lecteur aux trois ouvrages de base que sont [1], [2] et [3].

Nous voudrions mentionner enfin que ce travail s'inspire d'un certain nombre de documents, mais plus particulièrement de l'article de Glover [5].

2 - ALGORITHMES PROBABILISTES CONTROLES

La famille des "algorithmes probabilistes contrôlés", comme son nom l'indique (de façon plus ou moins nette), se caractérise par l'introduction volontaire et contrôlée des effets du hasard dans le déroulement du processus de résolution. Nous disons volontaire, car c'est dans le but bien précis de transcender le problème des optima locaux sans avoir à recourir à l'énumération qu'on introduit le hasard. Nous disons aussi contrôlée, car ces approches comportent des mécanismes restreignant l'impact du hasard à des situations prescrites et pour des durées limitées, assurant ainsi un comportement "globalement intelligent" du processus de résolution.

Cette stratégie a été proposée dès les années soixante comme un moyen pratique pour faire face à l'explosion combinatoire à laquelle sont sujet les problèmes NP-complets et cela en s'attaquant plus particulièrement à la question des optima locaux. Ainsi, en 1965, Reiter et Sherman [6] proposaient une nouvelle approche pour résoudre les problèmes d'optimisation combinatoire. L'idée de base de cette approche était de combiner des méthodes d'exploration "intelligente" (ou encore "dirigée"), comme en sont la plupart des heuristiques classiques, avec de l'exploration "au hasard". L'algorithme effectue de l'exploration intelligente tant qu'il le peut, puis il se tourne vers l'exploration "au hasard" pour éviter de rester coincé dans un optimum local.

On distingue généralement deux formes d'algorithmes probabilistes contrôlés selon l'utilisation qu'ils font de leurs composantes aléatoires

[5]. Ne connaissant pas de termes techniques officiels pour les désigner en français, nous avons pris la liberté de les identifier comme suit: les techniques de "points de départ aléatoires" (en anglais "random restart") et les méthodes de "brassage aléatoire" (en anglais "random shake-up"). Dans cette partie nous présenterons successivement chacune de ces deux stratégies ainsi qu'un raffinement de la seconde, le "recuit simulé". De plus, nous indiquerons les résultats obtenus lors de leur application à certains problèmes classiques d'optimisation combinatoire.

2.1 - Les techniques de points de départ aléatoires

2.1.1 Description et justification de la méthode

Les algorithmes qui souscrivent à cette approche sont nombreux, mais très semblables les uns aux autres. Ils se caractérisent par la superposition de techniques statistiques, notamment des techniques d'échantillonnage, à des heuristiques classiques d'amélioration itérative.

On y distingue les trois éléments suivants:

- une heuristique d'amélioration itérative (généralement du type "exploration locale");
- une procédure permettant de générer des solutions réalisables aléatoires;
- un critére d'arrêt global permettant de contrôler la procédure d'échantillonnage.

Le fonctionnement de ces algorithmes se schématise comme suit:

1. Générer une solution réalisable aléatoire.
2. Appliquer l'heuristique d'amélioration itérative à partir de cette solution "initiale" pour trouver de meilleures solutions; quand on ne peut plus faire d'améliorations, c'est qu'on se trouve en présence d'un optimum local (par rapport à l'heuristique utilisée) et on le garde en mémoire.
3. Vérifier si le critère d'arrêt global est atteint: si oui, on arrête le processus et on prend le meilleur des optima locaux trouvés en 2 comme "solution finale"; si non, on retourne à l'étape 1.

Chaque exécution de l'étape 2 du processus permet d'obtenir un optimum local. Quand cette opération est répétée plusieur fois à partir de points de départ différents, on effectue en quelque sorte une exploration plus ou moins "globale" du domaine réalisable. C'est là qu'on trouve la justification intuitive de cette approche: en effectuant ces diverses explorations, on augmente les chances de trouver un optimum global ou tout au moins un très "bon" optimum local.

D'un point de vue plus concret, lorsque l'algorithme s'arrête, on se trouve généralement en présence de plusieurs optima locaux distincts dont éventuellement l'optimum global. Si les points de départ ont été générés aléatoirement (c'est-à-dire selon une loi de probabilité uniforme) et qu'ils sont "suffisamment" nombreux, on peut alors considérer que la

distribution de l'échantillon d'optima locaux est une approximation de celle de la population-mère (les solutions réalisables) et que le meilleur de ces optima locaux est une bonne "estimation" de l'optimum global [7 et 8]. Or dans un contexte d'application pratique, l'assurance d'obtenir une "bonne" solution en un temps d'exécution raisonnable est bien souvent la seule chose qui compte, on comprend donc l'intérêt de cette approche.

2.1.2 Choix des composantes de l'algorithme

Le critère d'arrêt

La question principale qui survient dans le cadre de cette approche concerne le critère d'arrêt. En effet, celui-ci constitue le mécanisme de contrôle de la procédure et son choix à une influence déterminante sur la solution retenue, mais encore plus sur l'évaluation que l'on pourra faire de la qualité de cette solution. Dépendant de la connaissance plus ou moins restreinte du problème, des ressources disponibles et des objectifs visés, le choix de ce critère sera différent.

Le critère le plus usuel est le nombre de points de départ aléatoires. Dans le cas le plus simple, c'est un nombre k fixé a priori et "jugé" suffisant dans le contexte. La qualité de la solution ne peut alors être évaluée à proprement parler autrement que par comparaison avec d'autres solutions connues du même problème. Malgré sa "faiblesse" théorique cette option est assez attrayante dans un contexte où les ressources sont limitées. Une deuxième possibilité consiste à baser la

détermination du nombre k de points de départ aléatoires sur une analyse statistique préalable du comportement de l'heuristique d'amélioration itérative utilisée sur le problème étudié. C'est la méthode proposée par Lin [9] et reprise par Lin et Kernighan [7]. Cette analyse leur permet d'obtenir un estimé de la probabilité P_h que l'optimum local engendré par l'heuristique soit l'optimum global. On peut alors choisir k de façon à ce que l'expression $1 - (1 - P_h)^k$ qui représente la "probabilité" d'avoir obtenu l'optimum global, soit aussi proche de 1 qu'on le veut. On obtient ainsi un niveau de confiance associé à la solution obtenue.

D'autres critères d'arrêt plus sophistiqués que nous ne détaillerons pas ici (tels "le potentiel d'amélioration" ou "la valeur espérée du gain"), basés sur une estimation plus rigoureuse (du point de vue statistique) de la valeur de l'optimum global ont été proposés par Reiter et Sherman [6], Golden [10] et Lardinois [8].

L'heuristique d'amélioration itérative

Elle se doit d'être assez rapide puisqu'elle est appelée à être exécutée de façon répétitive. Comme nous l'avons indiqué plus tôt, elle est généralement du type exploration locale et nécessite donc la définition d'un voisinage (voir [1], chap.19). Ainsi, une heuristique "puissante" basée sur la définition d'un voisinage étendu sera susceptible de générer de meilleurs optima locaux qu'une autre dont le voisinage est plus restreint, mais ce gain de puissance se paie par une perte de rapidité, car il faut alors évaluer beaucoup plus de candidats. Son choix dépend donc lui

aussi du problème étudié et des objectifs visés. Notons aussi que plus une heuristique est "puissante", plus faible est alors le nombre d'optima locaux qu'elle engendre et élevée la fraction des points de départ aléatoires qui mènent à l'optimum global. On voit donc qu'il y a un compromis à faire entre la rapidité et la puissance: vaut-il mieux générer moins d'optimum locaux "forts" ou plus de "faibles"? La réponse à cette question varie selon le contexte et s'obtient empiriquement la plupart du temps.

Le générateur de points de départ aléatoires

Il n'y a pas grand chose à dire sur cet élément de l'approche autre qu'il doit générer de manière aléatoire des solutions réalisables pour le problème étudié. Dans certains cas, tel celui du problème du voyageur de commerce, l'obtention de solutions réalisables aléatoires est très facile: il suffit de choisir au hasard l'ordre de visite des sommets du graphe [7,9]. Dans d'autres contextes cette opération peut s'avérer beaucoup plus complexe et onéreuse (voir [1], chap.19).

Il faut aussi faire remarquer que, dépendant du problème étudié et de l'heuristique d'exploration locale choisie, la qualité des solutions initiales peut avoir un impact important sur les résultats obtenus (voir [1], Sec. 19.6). Dans cette optique, il est possible d'avoir recours à des algorithmes approximatifs du type "glouton" pour construire de bonnes solutions initiales. Mais pour que les points de départ ainsi générés le

soient aléatoirement, il faut prendre certaines mesures pour faire varier (de façon aléatoire) la procédure de construction [11].

2.1.3 Applications et résultats obtenus

L'approche des points de départ aléatoire à été appliquée à plusieurs problèmes classiques de l'optimisation combinatoire avec un succès certain. Nous présentons rapidement dans cette section quelques-unes de ces applications au Problème du voyageur de commerce (ou plus simplement du TSP - pour "Traveling Salesman Problem") et les résultats qu'elles ont produit. Rappelons tout d'abord la définition du problème: on doit visiter N villes, une et une seule fois, et revenir au point de départ tout en minimisant la distance parcourue.

Le premier article de référence est celui présenté par Lin [9] où les trois composantes de l'algorithme sont:

- l'heuristique d'exploration locale employée est basée sur des échanges du type "3-opt" (qui correspondent à éliminer trois arcs du circuit et à les remplacer par trois autres);
- le critère d'arrêt est le nombre de points de départ aléatoires nécessaires pour avoir une probabilité de 0,99 d'obtenir l'optimum global, tel que nous l'avons décrit plus tôt (c.f. page 9);
- les points de départ sont générés aléatoirement en tirant au hasard une séquence de visite des sommets.

Les résultats obtenus étaient extraordinaires pour l'époque (1965) et sont encore respectables aujourd'hui: on y résolvait des problèmes comptant environ 50 noeuds en 3 à 4 minutes de temps de calcul. L'extension de ces travaux, effectuée par Lin et Kernighan [7] et utilisant une heuristique d'échange plus sophistiquée (échanges du type "k-opt" où le nombre k d'arêtes échangées est variable), vint confirmer l'intérêt de l'approche. La taille des problèmes résolus passa à 100 noeuds pour des temps de calcul similaires (3 à 4 minutes). Il faut toutefois noter que, dans ces deux approches, les auteurs, dans le but d'accélérer le processus de résolution, font également appel à des techniques de "réduction" qui relèvent des stratégies d'apprentissage que nous verrons plus loin.

Même si l'observation précédente pouvait tempérer l'intérêt de cette approche, d'autres résultats obtenus par Golden [10] et plus encore une étude comparative empirique, citée par Golden et Stewart [12] et portant sur les principaux algorithmes approximatifs pour le TSP, sont venus confirmer ses qualités de rapidité et d'efficacité. Dans [10], on utilise l'algorithme de Clarke et Wright pour résoudre un problème de tournées (VRP) avec un seul véhicule de capacité infinie (donc un TSP). Notons que cet algorithme est de type "glouton", mais on génère des optima locaux aléatoirement en choisissant au hasard le noeud "origine" de la tournée. L'auteur y indique qu'il obtient des solutions qui se situent à environs 7% de l'optimum pour des problèmes comportant 130 noeuds et en des temps inférieurs à la minute. Dans [12], on peut voir qu'un des algorithmes de cette famille, soit l'heuristique "2-opt" exécutée à partir de 25 points de départ tirés au hasard, surpasse les approches de construction classiques

et rivalise avec les approches composites (i.e. celles comportant les trois phases habituelles d'initialisation, de construction et d'échange) tout en nécessitant moins de temps.

Mentionnons enfin que cette approche a aussi été appliquée avec succès au problème de partitionnement de graphes, voir Kernighan et Lin [13], et à un problème de design de réseau ("Minimum-cost survivable network problem"), voir Steiglitz et al. [11]. D'autres références sur les algorithmes probabilistes contrôlés, leurs applications et les aspects plus théoriques qui s'y rattachent sont disponibles dans [14].

2.2 Les méthodes de "brassage aléatoire"

2.2.1 Description et justification de la méthode générale

Dans cette deuxième famille d'algorithmes probabilistes contrôlés, au lieu d'appliquer la même heuristique à des points de départ différents, on essaye plutôt de générer des cheminements d'exploration distincts à partir du même point de départ ou d'un nombre restreint de ceux-ci. On obtient cet effet en interrompant périodiquement l'heuristique d'amélioration itérative de base pour générer des déplacements aléatoires, puis en l'appliquant à nouveau du point ainsi atteint. Cet artifice nous permet ainsi d'explorer des régions du domaine réalisable que l'heuristique n'aurait pas atteinte par elle-même [5].

On peut généralement distinguer cinq éléments dans la composition de ces méthodes:

- une procédure permettant d'obtenir une solution initiale réalisable;
- une heuristique classique (du type amélioration itérative le plus souvent);
- une phase aléatoire;
- une règle d'invocation de la phase aléatoire;
- un critère d'arrêt global de la procédure.

Dans son expression la plus répandue cette approche prend la forme suivante:

1. Générer une solution réalisable.
2. Appliquer l'heuristique d'amélioration itérative jusqu'à l'obtention d'un optimum local qu'on garde en mémoire.
3. Si le critère d'arrêt est atteint, arrêter la procédure et la "solution" fournie par l'algorithme est le meilleur des optima locaux obtenus jusque-là; sinon passer à l'étape 4.
4. Effectuer un certain nombre (prédéterminé ou variable) de déplacements aléatoires, puis retourner à l'étape 2.

Ici la règle d'invocation de la phase aléatoire est l'obtention d'un optimum local par l'algorithme d'exploration de l'étape 2.

Comme on peut le voir, le but de cette stratégie est encore une fois d'essayer d'atténuer les limitations des algorithmes classiques

d'exploration "dirigée" en ce qui a trait aux optima locaux, en induisant une exploration plus globale du domaine. En effet, quand on décide d'effectuer des déplacements aléatoires, on se dit qu'en acceptant ainsi des déplacements potentiellement "mauvais" du point de vue local, on pourra peut-être sortir de l'optimum local dans lequel on se trouve et reprendre la phase d'amélioration pour atteindre éventuellement une solution globalement plus attrayante. Cela correspond à essayer de remonter le long des "pentes" de la cuvette dans laquelle on se trouve, pour redescendre de l'autre côté. Dans le cadre d'un problème de minimisation, on peut schématiser le processus par un "saut" vers le haut (un genre de "back-tracking") suivi d'un déplacement latéral dans le but de reprendre la "descente" au-dessus d'une autre vallée que celle que l'on vient de quitter.

Des variantes de cette approche existent dans lesquelles c'est le cheminement même de l'heuristique d'exploration que l'on rend aléatoire en faisant varier les règles qui dirigent l'exploration en cours de route. Le recuit simulé que nous traiterons plus loin en est une.

2.2.2 Applications

Des méthodes de ce genre semblent avoir été appliquées à divers problèmes et dans des contextes variés. Malheureusement nous n'avons pu trouver aucune publication dans laquelle ce genre de techniques étaient évaluées de façon spécifique. Il nous est donc difficile de nous prononcer quant à leur efficacité par rapport aux méthodes classiques.

On peut toutefois mentionner que, dans Glover [5], une des variantes de cette approche est utilisée dans le cadre d'une stratégie d'apprentissage appliquée à un problème de confection d'horaires de production ("Job Shop Scheduling"). A diverses étapes de la résolution, l'algorithme utilisé choisit le critère de décision à appliquer en tirant celui-ci au hasard parmi un ensemble de candidats (selon une certaine distribution de probabilités).

Une autre application de ce genre de procédures se retrouve dans des contextes où des solutions initiales réalisables sont difficiles à trouver. On peut alors faire varier les critères de décision comme précédemment ou encore utiliser toujours le même critère, mais faire varier le cheminement en employant simultanément la règle d'acceptation de la "première amélioration rencontrée" et en ordonnant aléatoirement les candidats à évaluer dans le voisinage de la solution courante (voir [1], Section 19.6). En appliquant cette variante, il sera donc possible de générer des cheminements aléatoires à partir du même point de départ.

La dernière application que nous mentionnerons est celle, citée dans [14], d'utiliser ces approches probabilistes dans le but de surmonter les comportements pathologiques qui apparaissent dans les exemples du type "plus mauvais cas".

2.3 - Le recuit simulé

Comme le mentionne Glover [5], on peut voir le recuit simulé comme un raffinement (nous aurions plutôt tendance à dire une systématisation) de l'approche du "brassage aléatoire". En effet cette méthode de résolution comporte elle aussi une composante aléatoire qui influence le déroulement d'une procédure d'amélioration itérative, mais contrairement à l'approche classique du brassage aléatoire celle-ci peut être invoquée à chaque itération. L'aspect probabiliste est donc systématique dans la méthode.

2.3.1 Description et justification de la méthode

Le recuit simulé a été appliqué pour la première fois à des problèmes d'optimisation combinatoire par Kirkpatrick et al. [15]. Comme son nom le laisse entendre, il origine d'une analogie faite entre les problèmes d'optimisation combinatoire et le processus de "recuit" des corps solides en mécanique statistique. La technique du "recuit" consiste à chauffer suffisamment un matériau pour l'amener à l'état liquide, puis à le refroidir très lentement par paliers successifs (ceux-ci devant être suffisamment longs pour permettre au matériau d'y trouver son équilibre thermodynamique) pour atteindre en bout de ligne son état d'énergie minimale. Si le refroidissement s'effectue trop rapidement le matériau n'a pas le temps d'atteindre son équilibre aux divers paliers de température et des "défauts" (correspondant à des arrangements de particules sous-optimaux) apparaîtront dans la stucture finale. Celle-ci ne correspondra donc pas à celle de l'état d'énergie minimale.

On peut aisément transplanter tout cela dans le contexte de l'optimisation combinatoire en remarquant l'analogie qui existe entre le niveau d'énergie du matériau et la valeur de la fonction objectif qu'on cherche à minimiser. On peut alors considérer la recherche d'un équilibre à une température donnée comme la possibilité d'accepter des déplacements néfastes localement (c'est-à-dire qui font augmenter la valeur de la fonction objectif) dans le but de progresser éventuellement vers une configuration (lire une solution) globalement plus attrayante. Enfin la diminution progressive de la température dans le recuit peut se traduire par une propension décroissante de la méthode à accepter de "mauvais" déplacements à mesure que le temps passe et qu'on se rapproche de l'optimum global. Dans le cadre d'une procédure d'amélioration itérative, la "température" correspond à un paramètre permettant de contrôler la probabilité d'accepter un déplacement qui n'améliore pas la valeur de la fonction objectif.

Soit le problème d'optimisation combinatoire correspondant à minimiser la fonction $F(x)$ sur S où S est l'ensemble fini des états x que peut prendre un système, un algorithme de recuit simulé pour résoudre ce problème se caractérise par deux éléments:

1) une procédure permettant de choisir un successeur y à partir de l'état courant x (tout en simulant la recherche d'un équilibre telle que mentionnée ci-dessus);

2) un "schéma de refroidissement" permettant de gérer la succession des paliers.

L'algorithme généralement utilisé pour simuler ce processus de recherche d'équilibre est la méthode de Metropolis [16]. Celle-ci peut se schématiser comme suit:

1. A partir d'une solution réalisable x et d'un voisinage S_x on tire au hasard une solution y appartenant à S_x.
2. Si $F(y) \leq F(x)$, alors y remplace x comme solution courante et on retourne en 1.
3. Si $F(y) > F(x)$, alors on calcule $A_T(D) = \exp(-D/T)$, où $D = (F(y)-F(x))$, et on tire au hasard un nombre P compris entre 0 et 1. Si $P \leq A_T(D)$, alors on accepte le déplacement de x à y, on remplace x par y et on retourne à l'étape 1; sinon on garde x comme solution courante (c'est-à-dire on rejette le déplacement envisagé) et on retourne en 1.

Comme on peut le constater cet algorithme correspond à une procédure d'exploration locale tout ce qu'il y a de plus classique, mais dans laquelle on a introduit une composante aléatoire. Le paramètre T est le paramètre qui contrôle les probabilités d'acceptation. Il joue le même rôle que la température: pour des valeurs de T élevées les déplacements désavantageux seront acceptés souvent, alors que quand T tend vers 0 seuls les déplacements qui améliorent la fonction objectif seront susceptibles d'être retenus.

Sous certaines hypothèses (voir [17] et [18]), la séquence des états résultant de ces "déplacements" forme une chaîne de Markov homogène,

irréductible, finie et donc possédant une distribution d'équilibre [18]. Ceci implique que tous les états possibles du système, donc toutes les solutions réalisables du problème étudié, ont une probabilité non-nulle d'être visités lors de la marche aléatoire définie par l'algorithme de Metropolis (quand $T > 0$). Donc si on laisse le processus évoluer assez longtemps, l'exploration de S sera éventuellement complète avec probabilité égale à 1. De plus cette distribution d'équilibre a la forme d'une distribution de Boltzmann, ce qui correspond bien au comportement d'un système physique en équilibre (à une température donnée). Or une des propriétés de ce type de distribution est qu'au fur et à mesure que la température décroît, la distribution se concentre sur les états de basse énergie, puis, quand elle tend vers zéro, seul le ou les états d'énergie minimale ont une probabilité non-nulle d'exister. On voit donc facilement l'intérêt que représente cette approche pour régler le problème des optima locaux en optimisation combinatoire.

Malheureusement pour que le processus décrit précédemment converge sur un optimum global, il faut que le refroidissement soit effectué suffisamment lentement, c'est-à-dire en marquant suffisamment de paliers et en séjournant suffisamment longtemps dans chacun d'entre eux. Ce n'est déjà plus aussi simple. Nous arrivons ainsi à la définition du "schéma de refroidissement" qu'il faut suivre pour garantir que la procédure converge.

Dans le cadre d'un algorithme de recuit, un tel schéma de refroidissement se caractérise par: une liste établissant les paliers de température successifs (séquence décroissante des valeurs de T), ainsi que

le nombre de "déplacements" à effectuer à chacun de ces paliers avant de passer au suivant.

L'algorithme de recuit simulé se schématise alors comme suit:

1. Obtenir une solution initiale réalisable quelconque.
2. Pour chaque valeur de T prévue par le schéma de refroidissement, appliquer la méthode de Metropolis jusqu'à obtention du nombre de "déplacements" prescrits, puis passer à la température suivante.

Notons que l'algorithme ainsi défini ne s'arrêtera qu'à la fin du schéma de refroidissement, ce qui pourrait s'avérer très coûteux du point de vue temps d'exécution si le processus converge plus vite que prévu (ou qu'il reste bloqué assez tôt à cause d'un refroidissement trop rapide). Pour remédier à cette lacune potentielle, l'algorithme peut être équipé d'un critère d'arrêt supplémentaire, celui-ci pouvant prendre des formes plus ou moins compliquées (voir [15]).

2.3.2 Applications et résultats

Comme on peut le constater l'adaptation des concepts de la méthode du recuit simulé aux problèmes du domaine de l'optimisation combinatoire ne semble pas à première vue particulièrement ardue. Il suffit en effet de pouvoir définir pour le problème considéré [15]:

- une description concise des états du système;
- une fonction de coûts représentant bien la qualité de ces états;
- un mécanisme permettant de générer des déplacements (i.e. des transitions entre états "voisins");
- un schéma de refroidissement approprié.

En ce qui a trait aux trois premiers éléments de cette liste, ils peuvent généralement être obtenus sans trop de difficultés, le plus souvent à partir des heuristiques classiques disponibles pour traiter le problème considéré.

Pour ce qui est du dernier élément de l'approche, son choix est critique puisqu'il influe grandement sur l'efficacité de l'algorithme. En effet, si la température est abaissée trop brutalement l'algorithme convergera vers un optimum local pouvant être assez lointain de l'optimum global; par contre, si la température décroît trop lentement alors l'algorithme convergera bien vers l'optimum global mais le temps de calcul requis sera exorbitant. La plupart des auteurs s'entendent pour dire que la détermination d'un schéma de refroidissement efficace est un problème qui peut s'avérer assez complexe, long et onéreux. Kirkpatrick et al. [15] proposent de procéder par essai-et-erreur pour chaque type de problème, ou encore de partir d'une température suffisamment élevée pour être sûr que le système sera à l'état "fondu", puis de refroidir lentement par étapes (longues) jusqu'à ce que la "diffusion" cesse (i.e. que la fonction objectif ne varie plus). D'autres approches plus théoriques ont été proposées et font encore l'objet de recherches [20].

Voyons maintenant certains des problèmes qui ont été traités par des techniques de recuit simulé et les résultats ainsi obtenus.

Plusieurs applications de cette approche de résolution ont été réalisées pour le Problème du voyageur de commerce (TSP). Tout d'abord par Kirkpatrick et al. [15] qui ont illustré leur idée originale sur ce problème d'optimisation combinatoire. Ils utilisent comme mécanisme pour générer les transitions la méthode de Metropolis, où les "déplacements" (i.e. les transformations entre états voisins) sont définis par des transformations du type "2-opt" introduites par Lin [9]. Le schéma de refroidissement y est déterminé de façon empirique. Ils arrivent ainsi à obtenir des solutions localement optimales pour des instances de TSP dont la taille atteint les 6.000 sommets. De plus ils indiquent que la qualité de ces solutions est supérieure à celles fournies par l'approche heuristique plus classique qu'est la méthode de construction du "plus proche voisin" (en anglais "nearest neighbour"). Les auteurs ne mentionnent toutefois pas les temps de calcul requis pour obtenir ces solutions, non plus que le schéma de refroidissement employé ni la quantité de travail qu'a nécessité son élaboration. Il faut donc considérer ces résultats avec une certaine prudence avant d'en tirer des conclusions définitives.

Deux études plus poussées de la performance de cette méthode dans le contexte du TSP ont été publiées par Golden et Skiscim [21] et par Bonomi et Lutton [22], et obtiennent des résultats apparamment contradictoires. Pour toutes les deux, le mécanisme de transition est le même que celui utilisé par Kirkpatrick et al. [15], mais elles diffèrent dans le choix du

schéma de refroidissement. Dans [21] les résultats sont plutôt défavorables au recuit, celui-ci s'avérant moins rapide et moins précis que la méthode CCAO (pour une description de cette méthode voir [12]) et même qu'une approche du type "points de départ aléatoires" (c.f. section 2.1) employant des "2-opt" à partir de 25 points. Les auteurs obtiennent des résultats tout aussi décevants lors de l'application d'un algorithme similaire au Problème de localisation p-médiane. Les problèmes traités sont de taille moyenne, soit allant jusqu'à 100 sommets. Il faut toutefois noter, comme le font les auteurs, que le TSP est probablement le problème d'optimisation combinatoire qui a été le plus travaillé et donc pour lequel des heuristiques très performantes ont déjà été développées. Il n'est donc pas surprenant que la comparaison ne soit pas favorable au recuit, étant donné sa nouveauté et le peu d'expertise développée à date dans son ajustement fin.

Dans [22], par contre, les résultats sont beaucoup plus favorables au recuit, mais les auteurs ont quelque peu modifié l'algorithme utilisé. Ceux-ci commencent par limiter de façon astucieuse les échanges susceptibles d'être évaluées, en se basant sur la division du problème en sous-régions "géographiques" disjointes (c'est le principe bien connu du "divide-and-conquer"). De plus, ils partent d'une solution initiale de bonne qualité ce qui permet de commencer le recuit à une température beaucoup plus basse et donc d'effectuer un refroidissement plus lent en un temps moindre. La combinaison de ces deux modifications leur permet de réduire sensiblement le temps de calcul requis. Les problèmes traités ont des dimensions variant de 100 à 400 noeuds. Ils obtiennent des résultats où

le recuit "modifié" surpasse en précision une approche de "points de départ aléatoires" du même type que précédemment et une approche composée de l'heuristique de construction "convex hull" suivie d'une phase de "2-opt". De plus pour les problèmes dont la taille dépasse les 200 noeuds, leur algorithme de recuit est plus rapide que les deux autres approches.

Les mêmes auteurs reprennent avec succès cette même approche de recuit modifié pour traiter le Problème de couplage pondéré euclidien de poids minimal [18] et produire ainsi un algorithme très efficace, autant du point de vue qualité des solutions que de celui du temps de calcul. Bien que ce problème ne soit pas considéré comme un problème difficile puisqu'il existe un algorithme exact pour le résoudre en des temps de $O(N^3)$, mais pour des problèmes de grande taille ce temps devient exorbitant. L'algorithme de Lutton et Bonomi est très intéressant, car il permet d'obtenir des solutions très proches de l'optimum dans des temps très raisonnables.

Burkard et Rendl [19] ont adapté l'approche du recuit au Problème d'affectation quadratique et les résultats obtenus sont ici encore encourageants. Leur algorithme est de conception très simple (la plupart des paramètres sont choisis arbitrairement), mais ses performances sont comparables à celles d'autres algorithmes beaucoup plus sophistiqués ou réputés rapides.

Chams et al. [23], de leur côté ont expérimenté l'approche du recuit simulé sur le Problème du coloriage de graphes. Dans leur article ils

comparent les résultats obtenus avec trois algorithmes: une heuristique spécialisée, un algorithme de recuit simple (très semblable à celui employé dans [19]) et un algorithme "mixte" qui emploie d'abord l'heuristique pour construire rapidement une bonne solution partielle puis le recuit pour finir l'exploration. Les résultats obtenus sont à nouveau positifs: pour les graphes de moins de 200 noeuds le recuit est particulièrement efficace et demeure très performant pour ceux de plus grandes dimensions, mais c'est l'algorithme "mixte" qui s'avère le plus performant des trois.

Enfin, Greene et Supowit [17] ont appliqué la méthode du recuit à des Problèmes de partitionnement de graphes en obtenant de bons résultats. Mais ce qui est particulièrement intéressant dans leur article, c'est qu'ils y présentent une alternative à la méthode de Metropolis [16] pour générer les "déplacements". Cette variante qu'ils appellent "la méthode sans rejet" diffère de celle de Metropolis par la manière de choisir les "déplacements" ainsi que par la règle d'acceptation.

On peut la décrire comme suit:

Pour chaque déplacement possible i, pour $1 \leq i \leq N$, on garde en mémoire une valeur $w_i = a_T(D_i)$, où D_i est défini comme la variation de coût résultant du déplacement i et où $a_T(D_i) = \min(1, \exp(-D_i/T))$ (c'est l'équivalent du $A_T(D)$ de la section précédente); on choisit alors un déplacement i selon la probabilité $P = w_i / \sum_j w_j$, puis on effectue le déplacement en question et on met-à-jour les valeurs des w_i.

Comme on peut le constater, il n'y a plus de règle d'acceptation et donc plus de déplacements rejetés. De plus le choix des déplacements à effectuer n'est plus fait totalement au hasard, mais plutôt selon une distribution de probabilités qui reflète les conséquences du déplacement envisagé. Les auteurs montrent que la séquence d'états générés par la méthode "sans rejet" est équivalente en probabilité à celle de la méthode de Metropolis et qu'elle possède donc les mêmes propriétés de convergence.

L'intérêt de cette méthode "sans rejet" réside dans le fait que sa rapidité ne dépend pas de la température du système contrairement à celle de Metropolis. En effet, dans l'algorithme classique la rapidité d'exécution décroît à mesure que le taux d'acceptation diminue, or ce dernier est contrôlé par la température. Les auteurs, Greene et Supowit, montrent dans leur article que cette propriété de leur méthode se traduit par sa supériorité sur la méthode de recuit classique lorsque la température baisse en dessous d'une valeur critique. Ils suggèrent donc d'utiliser l'algorithme de recuit traditionnel jusqu'à ce que la température atteigne la valeur critique, puis de passer à la méthode "sans rejet". Une approche alternative qu'ils suggèrent également consiste à utiliser d'abord une heuristique pour construire une bonne solution, puis de passer directement à leur méthode en effectuant un recuit à partir d'une température assez basse.

Il faut toutefois remarquer que cette méthode alternative tire sa vitesse accrue d'une augmentation sensible de la mémoire qu'elle requiert. Quand les déplacements utilisés correspondent à des transformations assez

simples et surtout pas trop nombreuses, le compromis à faire entre les temps de calcul et la mémoire requise peut être très intéressant. Cependant, si les tranformations choisies sont complexes ou sujettes à une explosion combinatoire, il est peu probable que la méthode sans rejet se révèle attrayante.

3 - STRATÉGIES D'APPRENTISSAGE

La famille des stratégies d'apprentissage regroupe toutes les approches qui font appel à l'accumulation de "savoir" et à l'utilisation de cette "expérience" pour orienter et contrôler le processus de résolution. Les concepts qu'on y emploie sont très proches et bien souvent proviennent directement du domaine de l'Intelligence Artificielle. Ces techniques ont été appliquées à certains problèmes d'optimisation combinatoire dès les années soixante, mais n'ont pas donné suite à beacoup de recherches depuis. Elles ne constituent donc pas à proprement parler une approche de résolution concurrente aux autres, mais plutôt une alternative éventuelle qui, selon Glover [5], mérite d'être étudiée d'avantage.

3.1 Description et justification de la méthode

Le "savoir" auquel nous faisons référence peut prendre des formes très diverses comme la "forme" des solutions optimales, les performances d'une heuristique ou d'un critère de décision particulier lorsque confrontés à un type de problème ou de situation donnés, et bien d'autres... D'autre part l'accumulation et l'utilisation de ce savoir peut s'effectuer aussi bien de façon préalable à la résolution que concomitante à celle-ci ou mieux encore des deux manières.

Il existe donc une grande latitude dans la conception de ce genre d'approches ce qui rend malaisée leur description en termes précis. Il est

toutefois possible de les caractériser de façon plus ou moins grossière comme suit:

1. Déterminer à partir de l'expérience accumulée au préalable un ensemble de critères de décision initial et des règles d'invocation adaptées à ceux-ci. Combiner le tout pour définir un algorithme initial qu'on identifie alors comme l'algorithme "courant".
2. Effectuer une phase de résolution en appliquant l'algorithme "courant" au problème étudié un certain nombre de fois. On obtient ainsi un ensemble de résultats (solutions, performances et comportement de l'algorithme "courant").
3. Vérifier si le critère d'arrêt global de la procédure est atteint: si oui, on arrête; si non, on passe à l'étape 4.
4. Inférer, à partir des résultats obtenus en 2, des ajustements ou des modifications des règles d'invocation et éventuellement des critères de décision eux-mêmes (en abandonner certains, en concevoir de nouveaux, ...). Effectuer les ajustements jugés pertinents. On définit ainsi un nouvel algorithme qui remplace l'algorithme "courant", puis on retourne à l'étape 2.

Notons que dans la schématisation qui précède nous avons employé le terme "algorithme" à l'intérieur de la procédure de résolution elle-même car une stratégie d'apprentissage peut souvent se visualiser comme la superposition de techniques d'analyse et d'ajustement de paramètres à une structure algorithmique de base.

L'étape 4 est bien entendu l'étape critique de l'approche, le processus d'inférence qui s'y déroule influence grandement son succès. Il implique une analyse a posteriori des résultats obtenus dans le but de faire de la reconnaissance de formes ou de tendances (dans un sens large) et ainsi déterminer les aspects positifs ou négatifs de la version courante de l'algorithme. Une fois ceux-ci établis, il faut alors déterminer les correctifs à apporter. Or cette analyse et la conception des mécanismes d'ajustement peuvent ne pas être aisés à réaliser; ils constituent une des difficultés premières de ces approches.

Les approches relevant de cette famille ne trouvent pas leur justification dans des concepts mathématiques, ou autres, relevant de résultats "théoriques" plus ou moins complexes. Elles sont plutôt basées, comme bien des "algorithmes" d'Intelligence Artificielle, sur l'utilisation de beaucoup de règles très simples relevant souvent de notions de "bon sens". Leur avantage potentiel sur les méthodes classiques d'optimisation combinatoire provient de leur utilisation généralisée et astucieuse de sources d'informations variées auxquelles ces dernières ne font habituellement pas appel. Par exemple, le recours à divers "critères" de décision, bien que plus difficile à gérer, confère une grande flexibilité à la démarche de résolution ce qui permet d'espérer obtenir des solutions supérieures à celles fournies par une approche "rigide".

De plus ce genre de stratégie peut s'avérer d'un intérêt certain lorsque le problème considéré est peu ou pas connu. L'application d'une approche d'apprentissage permet alors d'explorer ce problème, d'accumuler

des connaissances à son sujet et d'obtenir en bout de ligne un algorithme adapté à ce type de problème alors qu'il n'en existait pas auparavant.

3.2 Applications et résultats

Ainsi que nous l'avons indiqué plus tôt, les publications traitant de ce sujet semblent être peu nombreuses. Dans son article [5], Glover décrit trois exemples de l'application de cette approche à des problèmes d'optimisation combinatoire: un problème d'ordonnancement des travaux dans un atelier (en anglais "job shop scheduling problem"), des problèmes de type voyageur de commerce et un problème de programmation par objectifs en nombres entiers ("integer goal programming problem"). Les deux premières correspondent à des travaux publiés antérieurement par le même auteur, mais dont il nous a été impossible de nous procurer des copies. Nous avons inclus leurs référence dans la bibliographie de ce travail à titre purement indicatif: ce sont respectivement les références [24] et [25].

Dans l'application au problème d'ordonnancement, Glover [5] décrit deux procédures de résolution. Dans la première, l'algorithme de base doit, à diverses étapes de la résolution, choisir selon une distribution de probabilités la "règle de décision" à appliquer parmi un ensemble de règles alternatives. L'approche consiste alors à résoudre le même problème à plusieurs reprises, on génère ainsi plusieurs solutions différentes (notons que c'est une application locale des principes des algorithmes probabilistes contrôlés dont nous avions parlé à la section 2). La procédure utilise ensuite ces solutions pour modifier les probabilités

associées aux règles choisies, de façon à favoriser celles qui ont mené à des solutions de bonne qualité (en augmentant leur probabilité d'être choisie) et en pénalisant celles qui se révélent mauvaises. La procédure fonctionne par phases successives de résolutions puis d'ajustements et elle "apprend" ainsi à améliorer ses performances.

Dans la deuxième application au problème d'ordonnancement, l'algorithme de base utilise une règle de décision composite dans laquelle sont "intégrés" tous les critères de décision concurrents de la version précédente. L'approche consiste alors à résoudre le problème plusieurs fois en faisant varier systématiquement les paramètres associés à chacun des critères de décision simples compris dans la règle composite. En effectuant le reste de l'approche de façon similaire à l'application précédente, la procédure ainsi définie réussit à "apprendre" à fixer les paramètres de manière appropriée. L'auteur déclare que les solutions obtenues par cette application étaient égales ou supérieures aux meilleures solutions connues pour les exemples traités.

Une approche en tous points semblable à la première des applications au problème d'ordonnancement a été présentée par Mabert et Whybark [28] pour un problème de localisation-allocation. Ici aussi l'approche procède par ajustements successifs d'une distribution de probabilités (associées aux noeuds candidats pour la localisation) et les résultats obtenus sont bons.

Lors de ces trois applications, la procédure d'ajustement des probabilités ou des paramètres se rapproche des concepts de "généralisation" et de "spécialisation" des règles de décision qu'on emploie en intelligence artificielle pour respectivement renforcer ou réduire l'applicabilité de ces règles (voir [26] et [27]).

L'application au problème du voyageur de commerce proposée par Glover [5] privilégie quant-à-elle un genre de reconnaissance des formes. La procédure utilise comme algorithme de base l'heuristique "2-opt" seule, qu'elle applique à partir d'un certain nombre de points de départ aléatoires. Puis elle analyse les (trois) meilleures solutions ainsi obtenues afin d'identifier les séquences de noeuds qui leur sont communes. Les arêtes correspondantes sont alors considérées comme "fixées", c'est-à-dire appartenant à la solution finale. La procédure reprend ensuite la phase de résolution, mais en travaillant sur un problème "réduit" puisque les arêtes fixées ne peuvent plus être impliquées dans des échanges. C'est la même technique qu'utilisent Lin [9] et Lin et Kernighan [7] (toujours sur le TSP) et qu'ils appellent "réduction". La motivation de cette démarche repose, comme l'indiquent les auteurs de [7], sur la présomption que ces arêtes sont apparues trop souvent dans les meilleures solutions pour que ce soit le fruit du hasard. Il est donc hautement probable qu'elles soient présentes dans l'optimum global, d'autant plus que celui-ci peut très bien être une des solutions déjà en main. Ces réductions du problème permettent des économies substantielles de temps de calcul.

Une extension de ce concept est présentée dans Glover [29] permettant ainsi de le généraliser à d'autres problèmes, mais aucune application n'y est présentée.

Enfin, toujours dans [5], Glover décrit l'application des concepts d'apprentissage à un problème du type "bin packing" (consistant à affecter des objets à des boîtes de façon à minimiser l'écart entre le "poids" résultant de l'affectation et des "poids cibles" associés aux boîtes), application qui s'est révélée pour le moins ardue. La procédure d'apprentissage était superposée à une heuristique constructive simple. La mise au point des mécanismes d'ajustement des règles de décision nécessita la combinaison de quatre concepts distincts avant de donner lieu à des améliorations significatives lors du passage d'une phase de résolution à la suivante, soit la preuve que l'algorithme était en train "d'apprendre". On voit donc que l'application d'une stratégie d'apprentissage peut, comme nous l'avions indiqué plus tôt, s'avérer difficile à réaliser et à mettre au point. D'ailleurs même dans le domaine de l'intelligence artificielle où l'on travaille depuis longtemps sur le concept d'apprentissage, les résultats obtenus dans le développement d'algorithmes capables d'apprendre réellement à partir de leurs expériences sont très limités [30]. Certains auteurs s'entendent même pour dire que le développement de tels algorithmes ne semble pas devoir s'effectuer dans un proche avenir [30,31].

Nous tenons à faire remarquer que cette mise en garde n'a pas pour but de dénigrer cette approche, mais seulement celui de souligner certaines des difficultés qu'elle peut présenter. En effet, ces dernières

considérations ne doivent pas faire oublier les succès obtenus lors de l'application de certains de ces concepts telle la réduction [5,7 et 9].

4 - RECHERCHE AVEC TABOUS

La "recherche avec tabous" est une approche de résolution toute neuve dans la gamme des outils de l'optimisation combinatoire. Elle a été présentée en tant qu'approche à part entière par Glover [5] en 1986, bien que l'idée de base remonte à 1977 (Glover [29]).

4.1 Description et justification de la méthode

La recherche avec tabous peut être vue comme une "méta-heuristique" qu'on superpose à une autre heuristique [5,32]. C'est une approche de résolution permettant de se déplacer pas-à-pas vers l'optimum global d'une fonction objectif, mais qui est munie de mécanismes visant à l'empêcher de se retrouver bloquée dans un optimum local.

Pour combattre le problème des optima locaux, la recherche avec tabous s'appuie sur une stratégie d'interdiction de certaines transformations de la solution courante (qu'elle déclare temporairement "tabou") dans le but d'empêcher la procédure de "cycler" lors de la résolution. Le cheminement s'effectue alors en identifiant à chaque itération la "meilleure" transformation possible sujette aux limitations imposées par la "liste de tabous". Celle-ci peut éventuellement entraîner une augmentation de la fonction économique (comme c'est également le cas dans le "recuit simulé") et c'est là que réside la justification de ces interdictions: elles permettent à l'exploration de se poursuivre sans retomber systématiquement dans les optima locaux desquels elle a réussi à

se sortir. On peut ainsi progresser au-delà des optima locaux, rencontrés en cours de route, vers un optimum global.

L'approche peut se schématiser comme suit [32]:

1. Trouver une solution initiale réalisable x et poser $T = \emptyset$ (c'est-à-dire qu'il n'y a aucun tabou).
2. Si l'ensemble, $S_x - T$, des transformations "permises" à partir de la solution courante x est vide, alors aller à l'étape 4. Sinon, on choisit parmi celles-ci la meilleure transformation possible "s" (élément de $S_x - T$), ce qui nous déplace de x à s(x).
3. On pose $x := s(x)$ (c.-à-d. on effectue le déplacement). De plus, si $f(s(x)) < f(x^*)$ (où x^* représente la meilleure solution obtenue jusqu'à présent), alors on pose $x^* := s(x)$.
4. Si le critère d'arrêt est vérifié, on arrête la procédure (c.-à-d. habituellement après un certain nombre d'itérations depuis le début ou depuis la dernière amélioration de x^*). Sinon, on effectue une mise à jour de T et on retourne à l'étape 2.

Notes: S_x est l'ensemble des transformations possibles lorsque la solution courante est "x".
T est l'ensemble des transformations considérées tabou à l'itération présente (i.e. la "liste des tabous").
f(x) est la fonction objectif du problème traité évaluée au point solution x.

Aprés cette description sommaire de la stratégie, il nous faut détailler et justifier certains de ses aspects.

Les Transformations

La détermination des transformations possibles à partir d'une solution donnée nécessite la définition d'une structure de voisinage pour le problème étudié. A chaque itération, les seules transformations considérées sont celles impliquant la solution courante et les éléments de son voisinage. L'évaluation des transformations "candidates" se fait selon le critère de la fonction objectif employée et, dans le contexte d'un problème de minimisation, la "meilleure" transformation est celle qui fait décroître le plus la valeur de la fonction objectif ou celle qui la fait augmenter le moins (advenant le cas où il n'y a aucune transformation permise qui amène une diminution de f).

Le choix de l'ensemble S_x des transformations candidates peut ne pas contenir toutes les transformations applicables à x. Notamment, quand la structure de voisinage utilisée est "grande", l'évaluation complète du voisinage consommerait certainement trop de ressources.Dans ce genre de situation, on se contente généralement de ne considérer à chaque itération qu'une portion de celui-ci, en générant aléatoirement un nombre limité de voisins de la solution courante et donc de transformations candidates.

Notons que l'utilisation du concept de "meilleure transformation disponible" plutôt que celui de "transformation améliorant la valeur de la fonction objectif" rend l'approche insensible au problème des optima locaux. En effet, on est toujours capable de définir la "meilleure" transformation, même quand il n'y a pas d'amélioration de cette fonction

objectif. L'exploration peut donc se poursuivre sans difficulté au-delà des optima locaux.

La liste de "tabous"

La définition et la gestion de la liste de tabous jouent bien entendu un rôle primordial dans l'approche. Ainsi cette liste correspond à une sorte de mémoire (à court terme) [33] qui indique à la procédure où elle a déjà été, lui permettant donc de diriger son exploration vers des régions du domaine non encore visitées.

A cet effet, les transformations inscrites dans cette liste ne sont pas les transformations effectuées auparavant, mais plutôt les opposées de celles-ci. On interdit ainsi la prise de toute décision qui nous ferait faire en sens inverse un chemin déjà exploré (pour éviter de retomber immédiatement dans un optimum local que l'on vient de quitter). Glover mentionne dans [5] et [32] que la représentation plus directe, correspondant à inscrire sur la liste les transformations précédentes dans le but d'empêcher leur répétition, ne s'est pas avérée une alternative adéquate.

D'autre part, pour conserver à l'approche suffisamment de flexibilité dans son exploration, le caractère tabou d'une transformation ne doit pas être maintenu en permanence (car cela aboutirait à une énumération complète). La liste des tabous contient donc un nombre limité "m" de transformations correspondant aux déplacements effectués lors des "m"

dernières itérations. La gestion de la liste s'effectue de façon cyclique: à chaque fois qu'on rajoute une nouvelle transformation interdite à la liste, on enlève de celle-ci la transformation qui y a résidé le plus longtemps.

La taille "m" de la liste peut être fixe ou variable dépendant de l'application et cette longueur représente en quelque sorte l'étendue de la mémoire que possède l'algorithme. Notons que les performances de l'approche sont fortement liées au choix de ce paramètre: une valeur suffisamment élevée de m permet de "garantir" qu'il n'y aura pas de cyclage, mais l'approche y perd en flexibilité et en rapidité lorsque comparée à celles où m prend des valeurs plus petites. Il y a un compromis à faire entre la sécurité et la flexibilité (synonyme ici de puissance) [32]. Dans son rapport sur la recherche avec tabous [32], Glover indique que contrairement aux résultats préliminaires mentionnés dans [5], la taille de la liste de tabous est fortement dépendante de celle du problème traité. Le choix d'une valeur de "m" adéquate est donc plus complexe qu'anticipé.

Nous mentionnons pour finir que si le problème traité l'exige, il est tout à fait possible de concevoir l'approche avec plusieurs listes de tabous permettant ainsi de différencier les transformations selon divers critères de classification (voir [5] et [32]).

Tabou conditionnel

Un raffinement de l'approche de base décrite plus haut est obtenu par l'introduction d'un mécanisme permettant de surmonter l'interdiction des tabous. Ainsi, dans le but d'augmenter la flexibilité de l'approche, on permet à des transformations inscrites sur la liste des tabous d'être considérées lors de l'étape 3, en autant qu'elles respectent certaines conditions. Plus spécifiquement, une transformation interdite peut perdre son statut tabou si son application à la solution courante permet une amélioration de la situation, sans pour autant introduire un risque de cyclage dans le processus.

Deux méthodes permettant d'arriver à cela sont proposées par Glover [5]. La première consiste à accepter de déroger à l'interdiction si l'application de s (où s est élément de T) à la solution courante x permet d'obtenir $f(s(x)) < f(x^*)$. C'est un critère très sévère qui ne devrait pas être vérifié souvent, il n'y aura donc pas grande flexibilité de plus puisque le statut tabou ne pourra presque jamais être renversé.

La deuxième méthode utilise une fonction **A(z)** déterminant le "niveau d'aspiration" associé à chaque valeur z de la fonction objectif. L'auteur propose de construire cette fonction comme suit:

1- Les valeurs z que prend la fonction objectif f(x) doivent être entières et varier de 1 à U, la valeur 0 constituant la borne inférieure de f(x) ;

2- On initialise en posant **A(z):= z-1** pour tout z;

3- Soient z' et z'' les valeurs de la fonction objectif lors de deux itérations successives, le niveau d'aspiration est mis à jour en posant simultanément **A(z'):= min(A(z'),z''-1)** et aussi **A(z''):= min(A(z''),z'-1)** .

Cette "mesure" constitue le but à atteindre par toute transformation appliquée à une solution courante dont la "valeur" est z. Elle correspond à la meilleure "amélioration" de ce type (i.e. obtenue en une seule itération à partir d'une solution courante x telle que f(x)= z) rencontrée jusqu'à présent, mais augmentée d'une unité.

Si une transformation étiquetée tabou permet d'améliorer la valeur de la fonction objectif de façon à la rendre égale ou inférieure à A(z) lorsqu'appliquée à la solution courante (dont la valeur est z), alors cette transformation peut être libérée de son statut. En effet, de par la construction de la fonction A(z), on est assuré qu'une telle transformation correspond à une transition du système qui ne s'est encore jamais produite, elle ne peut donc amener un cyclage éventuel de la procédure. Les deux conditions citées plus tôt sont ainsi respectées et l'interdiction peut être levée sans risque.

4.2 Applications et résultats

La méthode de "recherche avec tabous" est, comme nous l'avons

mentionné plus tôt, une stratégie de résolution récente ce qui explique le nombre relativement faible d'applications auxquelles elle a donné lieu.

Dans [34], Glover et McMillan utilisent une stratégie de recherche avec tabous conjointement avec des techniques de reconnaissance de formes et de décomposition pour résoudre un problème de confection d'horaires de travail ("general employee scheduling problem") pour les employés d'une chaîne de restauration rapide. L'horaire établi doit tenir compte de contraintes très diverses comme la disponibilité individuelle des employés, une convention collective,... La taille du problème traité compte plus de 1 million de variables (entières et réelles confondues) et de 3.000 à 9.000 contraintes. Les auteurs déclarent obtenir des solutions à 98% "optimales" en des temps intéressants avec un micro-ordinateur, mais ils ne fournissent pas de détails sur la façon dont la méthode est appliquée.

D'autres applications, où les concepts de base de la méthode étaient utilisés conjointement à d'autres stratégies de résolution, ont aussi été présentées dans [29] et [32]. Malheureusement aucune de ces applications mixtes ne permet d'évaluer la part de succès attribuable à chacun de ses éléments et donc de se prononcer sur le potentiel de la recherche avec tabous.

La première application de la méthode seule est réalisée dans Glover [5]. L'auteur l'applique à un problème consistant à affecter des charges à des canaux de façon à minimiser l'écart par rapport à des chargements "cibles". Les problèmes traités comportaient 144 variables binaires, 8

variables continues et 40 contraintes et peuvent se concevoir comme des problèmes de type "bin packing" multidimensionnels. Les transformations utilisée sont définies par des "échanges" d'objets (i.e. les charges à affecter) entre deux boîtes (i.e. les canaux à charger) et la liste des tabous est de longueur fixe m=7. L'auteur indique que les résultats obtenus sont très supérieurs à ceux obtenus par l'application d'une heuristique classique du type amélioration itérative basée sur les mêmes échanges. De plus, ces résultats s'avèrent également supérieurs à ceux produits par le logiciel commercial MPS spécialisé dans la résolution de problèmes de programmation mixte.

L'approche a été appliquée au problème du voyageur de commerce dans Knox et Glover [33] et Glover [32]. Les deux publications sont issues du même groupe de recherches et traitent donc le problème de la même manière. Dans leur application au TSP, les auteurs se basent sur l'heuristique d'exploration locale "2-opt" [9] pour définir les transformations considérées. Ils utilisent aussi la fonction de niveau d'aspiration A(z). Les résultats obtenus sur un ensemble de 7 problèmes dont les tailles variaient de 25 à 110 sommets se sont montrés très encourageants. Trois de ces problèmes, dont la solution optimale était connue, furent résolus optimalement en moins de 5 minutes (en moyenne) pour le plus gros (57 sommets) et cela sur un micro-ordinateur. Pour ce qui est des autres problèmes, la procédure obtint des solutions généralement supérieures aux meilleures solutions connues en des temps inférieurs à la minute (VAX 11/780), sauf pour le problème de 110 noeuds qui nécessita de 4 à 20 minutes dépendant du point de départ.

Cette application a surtout permis de constater que contrairement à ce qui était indiqué dans [5], la taille de la liste de tabous n'était pas indépendante de la dimension du problème traité. Les résultats obtenus à ce sujet par les auteurs leur suggèrent l'existence d'une relation linéaire entre la dimension du problème et la taille de la liste de tabous. Une dernière observation effectuée par les auteurs est que la solution initiale influence énormément les performances. Ainsi, dépendant de la solution initiale, la procédure peut aussi bien trouver la solution optimale très vite que cheminer longuement avant d'obtenir des résultats. Les causes de cette instabilité ne sont pas établies dans ces publications.

Dans [35], Hertz et de Werra présentent une application de cette stratégie au problème du coloriage de graphes. Ils utilisent l'approche telle que proposée par Glover [5], c'est à dire avec la fonction de niveau d'aspiration A(z), mais aussi avec une taille de liste de tabous fixe de m=7 (et donc indépendante de la dimension du problème). Les problèmes testés avaient des dimensions variant de 100 à 1.000 noeuds et les résultats étaient excellents. Les auteurs qui avaient publié auparavant une étude semblable utilisant la méthode du recuit simulé [23], indiquent que la méthode de recherche avec tabous obtient de très bons résultats. Ils mentionnent que les résultats de cette approche sont de qualité généralement supérieure à ceux du recuit et qu'ils nécessitent moins de temps de calcul. Enfin, comme dans [23], les auteurs indiquent qu'ils obtiennent les meilleurs résultats en appliquant une méthode mixte constituée d'une bonne heuristique de construction dans une première phase, suivie de l'application de la recherche avec tabous.

5 - CRITIQUE ET CONCLUSION

Dans les sections précédentes nous avons décrit trois des approches qu'identifiait Glover dans son article sur les nouvelles approches de résolution pour les problèmes d'optimisation combinatoire [5]. Nous avons pu constater que ces approches ne correspondent pas vraiment à l'image conventionnelle d'une heuristique, mais s'apparentent plutôt à des stratégies de contrôle permettant de "diriger" des heuristiques classiques vers un optimum global et cela malgré la présence d'optima locaux.

Ainsi, chacune de ces approches correspond à la mise en oeuvre d'un principe de résolution particulier devant permettre d'atteindre ce but:

- dans le cas des algorithmes probabilistes contrôlés, on distingue deux principes soit celui de "l'échantillonnage" qu'on retrouve dans les méthodes de points de départ aléatoires, et celui de "l'incohérence contrôlée de l'exploration" qu'on emploie dans les techniques de brassage aléatoire, dont le recuit simulé, lorsqu'on permet d'effectuer des déplacements qui dégradent localement l'état du système;

- dans les stratégies d'apprentissage le principe directeur est "l'accumulation d'expérience" et c'est cette expérience qui sert à diriger l'exploration;

- et enfin dans la recherche avec tabous, c'est l'utilisation du principe "d'empêcher de cycler" combinée à la définition d'un "meilleur déplacement disponible" qui permet à l'exploration de se poursuivre au-delà des optima locaux rencontrés.

Chacune de ces approches a obtenu des succès intéressants (voire même impressionnants quelques fois) lors de leurs applications à un certain nombre de problèmes d'optimisation classiques. Toutefois, toutes ces approches ont également montré lors de ces études qu'elles ne sont pas infaillibles et que les principes qui les supportent présentent des limitations.

L'approche des points de départ aléatoires est sans doute la plus facile à appliquer puisqu'elle ne nécessite pas d'ajustement complexe de paramètres. Elle est très efficace pour des problèmes de tailles petite à moyenne (TSP où N<100) et elle se prête facilement à une évaluation rigoureuse (statistiquement) de la qualité de la solution fournie. Toutefois, quand la dimension du problème traité devient grande (TSP où N>200 selon [22]) le principe de "l'échantillonnage" s'avère souvent inapplicable (trop coûteux), car il implique un nombre élevé de points de départ pour pouvoir engendrer une exploration adéquate du domaine des solutions réalisables. De plus, étant donné le principe même de l'approche, il y a une grande quantité de travail inutile puisqu'à chaque nouvelle phase d'exploration (à partir d'un nouveau point de départ) toute l'information accumulée lors de la phase précédente est perdue.

De leur côté, les principes directeurs des méthodes de brassage aléatoire et de la recherche avec tabous permettent aussi une adaptation assez facile de ces approches à la plupart des problèmes. Toutefois, celles-ci requièrent l'ajustement de certains paramètres, opération qui peut s'avérer fort complexe. Cependant, ces méthodes semblent mieux

supporter l'explosion combinatoire qui accompagne les problèmes de grande taille. De plus, elles ont l'avantage de procéder à une exploration ininterrompue (ou dans le cas du "brassage" classique presque ininterrompue) puisqu'elles ne sont pas stoppées par les optima locaux. On évite donc une partie du travail "inutile" présent dans la méthode des points de départ aléatoires. Toutefois dans le cas du brassage aléatoire (incluant le recuit simulé), la nature aléatoire des "mauvais" déplacements qu'on accepte peut diminuer singulièrement l'efficacité de l'exploration en laissant place à de nombreuses répétitions et donc ralentir considérablement le processus. Par opposition à cela, dans la recherche avec tabous le choix du déplacement à effectuer n'est plus aléatoire, mais correspond à la meilleure alternative disponible, sous réserve qu'elle ne nous ramène pas vers des solutions déjà visitées. On élimine ainsi les répétitions inutiles.

Enfin, l'accumulation d'expérience (ou de savoir) que constitue l'apprentissage est évidemment un concept très puissant, puisqu'il sous-entend une certaine forme d'intelligence. Des applications limitées de ces techniques ont fourni des résultats intéressants. Toutefois une application généralisée de ce concept semble pour l'instant hors de portée étant donné les difficultés importantes de modélisation qu'elle présente [30,31].

On peut donc établir une sorte de hiérarchie sommaire entre ces approches selon leur "puissance". La méthode des points de départ aléatoires est la plus simple de ces nouvelles approches et vraisemblablement la moins puissante d'entre elles. A l'autre extrême, on

retrouve l'apprentissage . Le recuit simulé et les méthodes de brassage aléatoire en général, ainsi que la recherche avec tabous se situent quelque part entre les deux. Il n'est donc pas étonnant que ce soient le recuit et la recherche avec tabous qui semblent être les approches les plus prometteuses dans l'état actuel des connaissances. Celles-ci paraissent en effet présenter un meilleur compromis entre la puissance et la complexité que les deux autres.

Une autre observation qu'il nous semble utile de souligner c'est les résultats très encourageants obtenus par ce que nous avons à quelques reprises identifié comme des "approches mixtes". Celles-ci correspondent à l'application d'une heuristique simple pour d'abord construire rapidement une bonne solution (ou une base de bonne solution), puis à poursuivre la résolution en utilisant une approche de recuit ou de recherche avec tabous [23,35]. Cette stratégie permet alors de tirer parti des avantages des deux méthodes, à savoir respectivement la vitesse et la puissance, tout en évitant (du moins partiellement) leurs défauts respectifs, soit l'arrêt dans le premier optimum local et la lenteur potentielle.

Cette notion d'approche mixte peut être étendue à de nombreuses autres approches hybrides combinant aussi bien ces nouveaux principes de résolution entre eux qu'avec d'autres plus classiques, telle la décomposition [34]. Ceci ouvre donc un vaste champ de recherches à exploiter.

RÉFÉRENCES

[1] Papadimitriou, C.H. et Steiglitz, K., Combinatorial Optimization : Algorithms and Complexity, Prentice Hall, New York, 1982.

[2] Salkin, H.M., Integer Programming, Addison-Wesley, Reading, Mass., 1975.

[3] Sakarovitch, M., Optimisation Combinatoire - Programmation Discrète, Hermann, Paris, 1984.

[4] Crowder, H., Johnson, E.L. et Padberg, M., "Solving Large-Scale Zero-One Programming Problems", Operations Research, Vol.31, No.5, 1983, pp.803-834.

[5] Glover, F., "Future Paths for Integer Programming and Links to Artificial Intelligence", Computers and Operations Research, Vol.13, No.5, 1986, pp.533-549.

[6] Reiter, S. et Sherman, G., "Discrete Optimizing", J. Soc. Indust. Appl. Math., Vol.13, No.3, 1965, pp.864-889.

[7] Lin, S. et Kernighan, B.W., "An Effective Heuristic for the Traveling Salesman Problem", Operations Research, Vol.21, 1973, pp.498-516.

[8] Lardinois, C., "Méthodes Statistiques en Optimisation Combinatoire", Publication No.116, Centre de recherche sur les transports, Université de Montréal, 1979.

[9] Lin, S., "Computer Solutions of the Traveling Salesman Problem", Bell System Tech. J., Vol.44, 1965, pp.2245-2269.

[10] Golden, B.L., "A Statistical Approach to the TSP", Networks, Vol.7, 1977, pp.209-225.

[11] Steiglitz, K., Weiner, P. et Kleitman, D.J., "The Design of Minimal Cost Survivable Networks", IEEE Trans. Circuits and Systems, Vol. CT-16, 1969, pp.455-460.

[12] Golden, B.L. et Stewart, W.R., "Empirical Analysis of Heuristics", dans The Traveling Salesman Problem, Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. et Shmoys, D.B., éditeurs, John Wiley & sons, Chichester, 1985.

[13] Kernighan, B.W. et Lin, S., "An Efficient Heuristic Procedure for Partitioning Graphs", Bell System Tech. J., Vol.49, 1970, pp.291-307.

[14] Maffioli, F., Speranza, M.G. et Vercellis, C., "Randomized Algorithms", dans Combinatorial Optimization : Annotated Bibliographies, O'hEigeartaigh, M., Lenstra, J.K. et Rinnooy Kan, A.H.G., éditeurs, Center for Mathematics and Computer Sciences, Amsterdam et John Wiley & sons, 1987.

[15] Kirkpatrick, S., Gelatt, C.D.Jr. et Vecchi, M.P., "Optimi-zation by Simulated Annealing", Science, Vol.220, No.4598, 1983, pp.671-680.

[16] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. et Teller, E., "Equation of State Calculations by Fast Computing Machines", J. of Chemical Physics, Vol.21, No.6, 1953, pp.1087-1091.

[17] Greene, J.W. et Supowit, K.J., "Simulated Annealing Without Rejected Moves", IEEE Trans. Comp. Aided Design, Vol.CAD-5, No.1, 1986, pp.221-228.

[18] Lutton, J.L. et Bonomi, E., "Simulated Annealing Algorithm for the Minimum Weighted Perfect Euclidean Matching Problem", RAIRO Recherche Opérationnelle, Vol.20, No.3, 1986, pp.177-197.

[19] Burkard, R.E. et Rendl, F., "A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems", European Journal of Operations Research, Vol.17, 1984, pp.169-174.

[20] Hajek, B., "Cooling Schedules for Optimal Annealing", Report, Dept. of Elec. Eng., Univ. of Illinois at Urbana-Champaign, 1986, submited to Math. of Oper. Res..

[21] Golden, B.L. et Skiscim, C.C., "Using Simulated Annealing to Solve Routing and Location Problems", Naval Research Logistics Quarterly, Vol.33, 1986, pp.261-279.

[22] Bonomi, E. et Lutton, J.L., "The N-City Traveling Salesman Problem: Statistical Mechanics and the Metropolis Algorithm", SIAM Review, Vol.26, No.4, 1984, pp.551-568.

[23] Chams, M., Hertz, A. et de Werra, D., "Some Experiments with Simulated Annealing for Coloring Graphs", European Journal of Operations Research, Vol.32, 1987, pp.260-266.

[24] Crowston, W., Glover, F., Thompson, G. et Trwick, J., "Probabilistic and Parametric Learning Methods for the Job Shop Scheduling Problem", GSIA, Carnegie Institute of Technology, 1964.

[25] Glover, F., "A Strategy for Traveling Salesman Problems", GSIA, Carnegie Institute of Technology, 1964.

[26] Charniak, E. et McDermott, D., Introduction to Artificial Intelligence, Addison-Wesley, Reading, Mass., 1985.

[27] Rich, E., Artificial Intelligence, McGraw-Hill, New York, 1983.

[28] Mabert, V.A. et Whybark, D.C., "Sampling as a Solution Methodology", Decision Sciences, Vol.8, 1977, pp.167-179.

[29] Glover, F., "Heuristics for Integer Programming Using Surrogate Constraints", Decision Sciences, Vol.8, 1977, pp.156-166.

[30] Lenat, D.B., "Computer Software for Intelligent Systems", Scientific American, Vol.251, No.3, 1984, pp.204-213.

[31] Waldrop, M.M., "Artificial Intelligence (I): Into the World", Science, Vol.223, pp.802-805.

[32] Glover, F., "Tabu Search", Report 88-3, Center for Applied Artificial Intelligence (CAAI), Graduate School of Business, University of Colorado, Boulder, Col., 1988.

[33] Knox, J. et Glover, F., "Tabu Search: An Effective Heuristic for Combinatorial Optimization Problems", article présenté à la Rocky Mountain Conference on Artificial Intelligence, 1988.

[34] Glover, F. et McMillan, C., "The General Employee Scheduling Problem: An Integration of MS and AI", Comput. & Oper. Res., Vol.13, No.5, 1986, pp.563-573.

[35] Hertz, A. et de Werra, D., "Using Tabu Search for Graph Coloring", Computing, Vol.39, 1987, pp.345-351.